DETAILED ACTION

1. Claims 1, 3 – 22, 24 – 43 and 45 – 53 are pending in the application. By this examiner's amendment, claims 1, 10, 22, 29, 32, 43 and 47 have been amended and claims 6, 19, 28 and 40 are cancelled.

INTERVIEW SUMMARY

2. During a telephone interview with Mr. Robert C. Kowert on July 1, 2008, examiner indicated the claims would be allowable if amended to incorporate the features of dependent claims 6, 19, 28, 29 or 40. Mr. Kowert agreed with the examiner's suggestion and authorized examiner to make the changes in an examiner's amendment.

EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Robert C. Kowert on July 1, 2008.

The application has been amended as follows:

a. Cancel claims 6, 19, 28, 40;

b. Claim 29, line 1, replace "claim 28" with --claim 22--;

c. Replace claims 1, 10, 22, 32, 43 and 47 with the following:

1. A computer-implemented method for the secure exchange of objects in a distributed computing environment, comprising:

a user accessing a client device;

generating a computer programming language object from a data markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device;

wherein said generating a computer programming language object from a data markup language representation of the object is performed by a virtual machine executing within the client device;

the client device receiving a message in the data markup language from a service device in the distributed computing environment prior to said generating a computer programming language object, wherein the message includes the data markup language representation of the object;

the user terminating said accessing the client device; and

deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

10. A computer-implemented method for the secure exchange of objects in a distributed computing environment, comprising:

a user accessing a client device;

the client device receiving a message in a data markup language from a service device in the distributed computing environment, wherein the message includes a data markup language representation of a computer programming language object;

determining if the user has access rights to the computer programming language object;

if said determining determines the user has access rights to the computer programming language object, generating the object from the data markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device; and

if said determining determines the user does not have access rights to the computer programming language object, not generating the object;

wherein said generating a computer programming language object from a data markup language representation of the object is performed by a virtual machine executing within the client device.

22. A device, comprising:

a processor; and

a memory coupled to the processor, wherein the memory stores program instructions executable by the processor to:

accept user input to initiate user access of the device;

generate a computer programming language object from a data markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the device;

implementing a virtual machine, wherein generating the computer programming language object is performed by the virtual machine;

receive a message in the data markup language from a source prior to said generating a computer programming language object, wherein the message includes the data markup language representation of the object;

terminate said user access; and

delete the computer programming language object in response to said terminating access;

wherein the deleted object is not accessible for use by subsequent users of the device.

32. A distributed computing system, comprising:

a client hardware device; and

a service hardware device:

wherein the client hardware device is configured to:

accept user input to initiate user access of the device;

receive a message in a data markup language from the service hardware device, wherein the message includes a data markup language representation of a computer programming language object;

determine if the user has access rights to the computer programming language object;

if said determining determines the user has access rights to the computer programming language object, generate the object from the data markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible far use during said accessing the client hardware device; and

if said determining determines the user does not have access rights to the computer programming language object, not generate the object;

wherein the client hardware device, is further configured to execute a virtual machine, wherein generating the computer programming language object is performed by the virtual machine.

43. A computer accessible storage medium storing program instructions, wherein the program instructions are computer-executable to implement:

a user accessing a client device;

generating a computer programming 1mguagc object from a data markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device;

wherein said generating a computer programming language object from a data markup language representation of the object is performed by a virtual machine executing within the client device;

receiving a message in the data markup language from a service device in the distributed computing environment prior to said generating a computer programming language object, wherein the message includes the data markup language representation of the object;

the user terminating said accessing the client device; and

deleting the computer programming language object in response to said terminating access, wherein the deleted object is not accessible for use by subsequent users of the client device.

47. A computer accessible storage medium storing program instructions, wherein the program instructions are computer-executable to implement:

a user accessing a client device;

the client device receiving a message in a data markup language from a service device in the distributed computing environment, wherein the message includes a data markup language representation of a computer programming language object;

determining if the user has access rights to the computer programming language object;

if said determining determines the user has access rights to the computer programming language object, generating the object from the data markup language representation of the object, wherein the object is an instance of a class in the computer programming language, and wherein the object is accessible for use during said accessing the client device; and

if said determining determines the user does not have access rights to the computer programming language object, not generating the object;

wherein said generating a computer programming language object from a data markup language representation of the object is performed by a virtual machine executing within the client device.

CONTACT INFORMATION

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Li B. Zhen
Primary Examiner
Art Unit 2194

/Li B. Zhen/
Primary Examiner, Art Unit 2194